

The benefits of FPGA coprocessing

By Tom Hill

The Xilinx ESL Initiative brings the horsepower of an FPGA coprocessor closer to the reach of traditional DSP system designers.

High-performance DSP platforms, traditionally based on general-purpose DSP processors running algorithms developed in C, have been migrating towards the use of an FPGA preprocessor or coprocessor. Doing so can provide significant performance, power, and cost advantages (see Figure 1).

Even with these considerable advantages, design teams accustomed to working on processor-based systems may avoid using FPGAs because they lack the hardware skills necessary to use one as a coprocessor. Unfamiliarity with traditional hardware design methodologies such as VHDL and Verilog limits or prevents the using an FPGA, oftentimes resulting in more expensive and power-hungry designs. In this issue, Tom explains how a new group of emerging design tools called ESL (electronic system level) promises to address this methodology issue, allowing processor-based developers to accelerate their designs with programmable logic while maintaining a common design methodology for hardware and software.

Boosting performance with FPGA coprocessing

Designers can realize significant improvements in the performance of a DSP system by taking advantage of the flexibility of the FPGA fabric for operations benefiting from parallelism. Common examples include (but are not limited to) FIR filtering, FFTs, digital down conversion, and forward error correction (FEC) blocks. Xilinx Virtex-4 and Virtex-5 architectures provide as many as 512 parallel multipliers capable of running in excess of 500 MHz to provide a peak DSP performance of 256 GMACs. By offloading operations that require high-speed parallel processing onto the FPGA and leaving operations that require high-speed serial processing on the DSP, the performance and cost of the DSP system are optimized while lowering system power requirements.

But FPGA embedded processing can also lower costs. A DSP hardware system that includes an FPGA coprocessor offers numerous implementation options for the operations contained within the C algorithm, such as partitioning the algorithm between the DSP processor, the

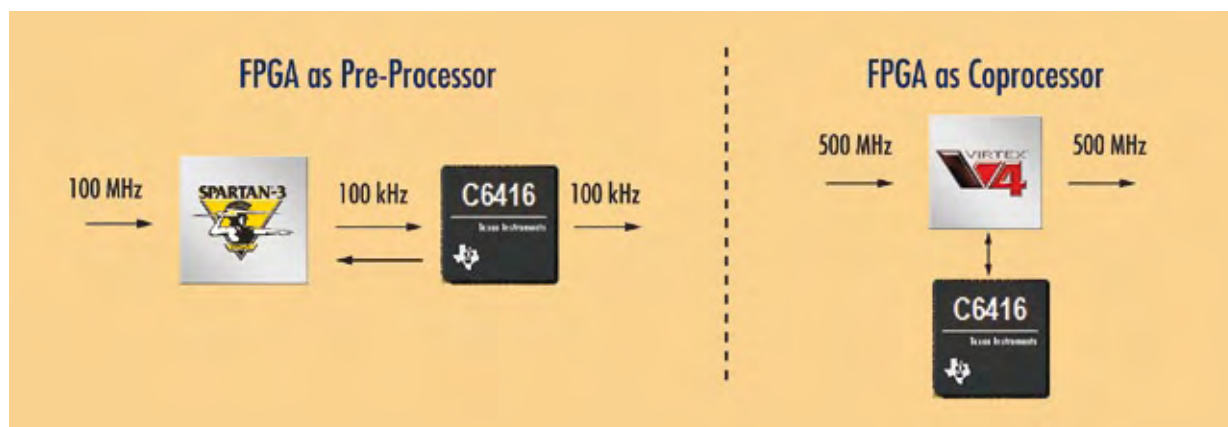


Figure 1

FPGA-configurable logic blocks (CLBs), and the FPGA embedded processor. The Virtex-4 device offers two types of embedded processors: the MicroBlaze soft-core processor, often used for system control, and the higher performance PowerPC hard-core embedded processor. Parallel operations partitioned into the FPGA fabric can be used directly in a DSP datapath or configured as a hardware accelerator to one of these embedded processors.

The challenge facing designers is how to partition DSP system operations into the available hardware resources in the most efficient and cost-effective manner. How best to use FPGA embedded processors is not always obvious, but this hardware resource can have the greatest impact on lowering overall system cost. FPGA embedded processors provide an opportunity to consolidate all noncritical operations into software running on the embedded processors, minimizing the total amount of hardware resources required for the system.

From C to gates

When targeting an FPGA, the term *C to gates* refers to a C-synthesis design flow that creates one of two implementation options – direct implementation onto the FPGA fabric as a DSP module, or the creation of a hardware accelerator for use with the MicroBlaze or PowerPC 405 embedded processor (shown in Figure 2). When an operation lies directly in the DSP datapath, the highest performance is achieved by implementing an operation as a DSP module. This involves synthesizing the C code directly into RTL and then instantiating the block into the DSP datapath. Designers can perform this instantiation using traditional HDL design methodologies or through system-level design tools such as System Generator for DSP. Through direct instantiation, the highest performance can be achieved with minimal overhead.

The leading C-synthesis tools are capable of delivering performance approaching that of hand-coded RTL – but achieving this requires detailed knowledge of C-synthesis tool operation and coding styles. Code modifications and the addition of in-line synthesis instructions for inserting parallelism and pipeline stages are typically required to achieve the desired performance. Even with these modifications, however, the productivity gains can be significant. The C-system model remains the golden source driving the design flow.

An alternative and often simpler approach is to create a hardware accelerator for one of the Xilinx embedded processors. The processor remains the primary target for the C routines, with the exception that performance-critical operations are pushed to the FPGA logic in the form of a hardware accelerator. This provides a more software-centric design methodology. However, some performance is sacrificed with this approach. C routines are synthesized to RTL, similar to the DSP module approach, except that the top-level entity is wrapped with interface logic to allow it to connect to one of the Xilinx embedded processor buses. This creates a hardware accelerator that can be imported into the Xilinx EDK environment and called through a software-friendly C function call.

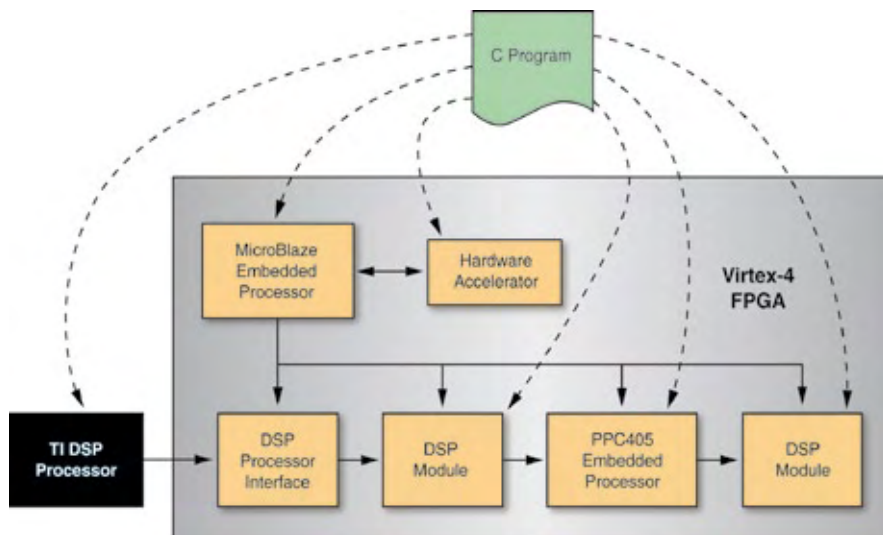


Figure 2

DSP chip selection criteria

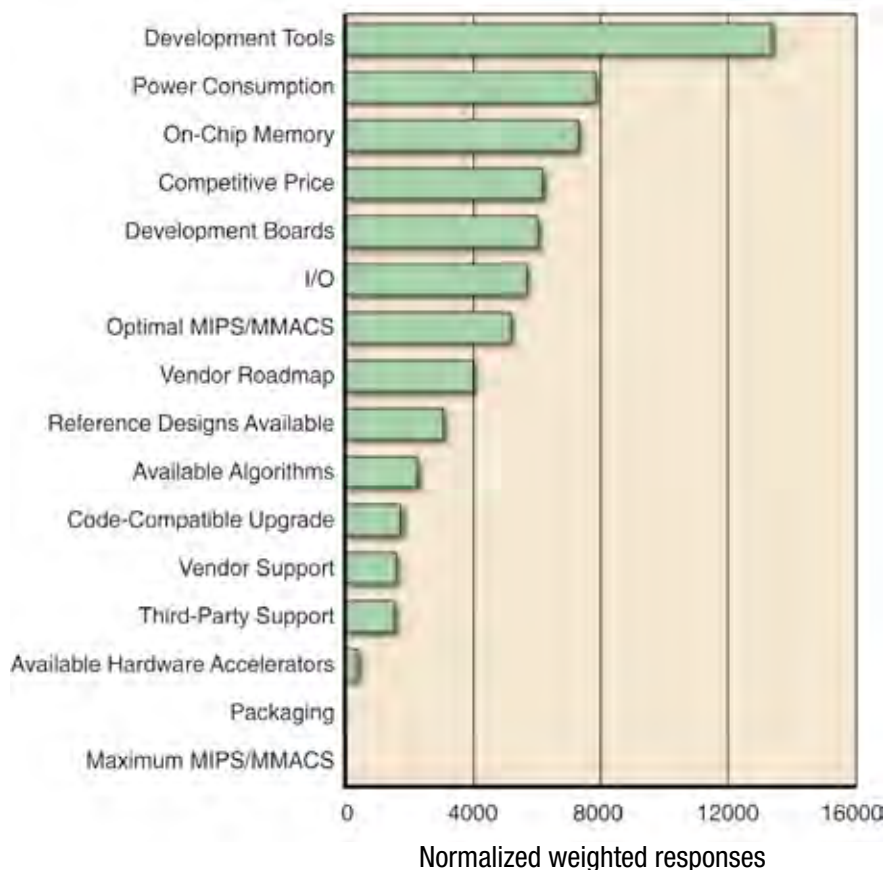


Figure 3

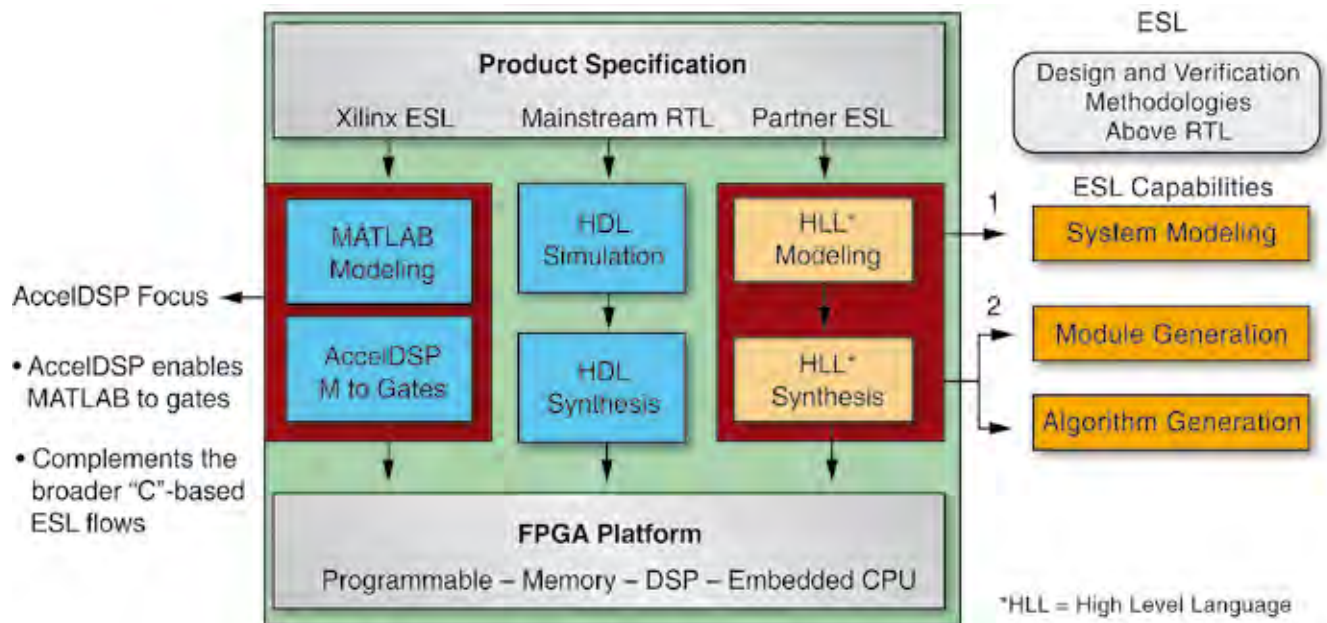


Figure 4

The performance requirements for mapping C routines into hardware accelerators are typically less aggressive. Here the objective is to accelerate the performance beyond that of pure software while maintaining a software-friendly design flow. Although the coding techniques and in-line synthesis instructions are still available, you can typically achieve your desired performance gains without their use.

Design methodology – the barrier to adoption

The effort and breadth of skill required to correctly partition and implement a complex DSP system is formidable. In 2005, the market research firm Forward Concepts conducted a survey to determine the most important FPGA selection criteria for DSP. The published results, shown in Figure 3, identify development tools as the most important. [Editor's note: Will Strauss from Forward Concepts writes the Forward Thinking column for DSP-FPGA.com.]

The survey illustrates that the benefits of a DSP hardware system utilizing an FPGA coprocessor are well understood, but that the current state of development tools represents a significant barrier to adoption for traditional DSP designers.

The Xilinx ESL Initiative

ESL design tools are pushing digital design abstraction beyond RTL. A subset of these tool vendors are specifically focused on mapping system models developed in C/C++ into DSP hardware systems that include FPGAs and DSP processors. Their vision is to make the hardware platform transparent to the software developer (see Figure 4).

Rather than attempting to solve one piece of this methodology puzzle internally, this year the company launched a partnership program with key providers of ESL tools called the *ESL Initiative*. The focus of this partnership is to empower designers with software programming skills to be able to easily implement their ideas in programmable hardware without having to learn traditional hardware design skills. This program is designed to accelerate the development and adoption of world-class design methodologies through innovation within the ESL community.

When combined, the collective offerings from Xilinx ESL partners offer a wide spectrum of complementary solutions that are optimized for a range of applications, platforms, and end users. Xilinx too has focused its efforts on complementary technology. For example, AccelDSP Synthesis provides a hardware path for algorithms developed in floating-point MATLAB, while System Generator for DSP allows modules developed using ESL designs to be easily combined with Xilinx IP and embedded processors. The quickest path to realizing a programmer-friendly FPGA design flow is through a motivated and innovative set of partners.

For more information about the ESL Initiative, visit www.xilinx.com/esl. DSP-FPGA.com

Tom Hill has more than fourteen years experience in the EDA industry. Hill oversees all product, strategic and corporate marketing activities related to the Xilinx development tool



System Generator for DSP. Hill was most recently at AccelChip Inc., where he was technical marketing manager responsible for product direction and application of high-level design methodologies and tools for DSP applications. Prior to AccelChip he held positions as product manager, technical marketing manager, technical marketing engineer and field applications engineer for various FPGA and ASIC synthesis tools companies. Hill began his career as a hardware and ASIC design engineer at Allen-Bradley and Lockheed. Tom Hill holds a B.S. degree in electrical engineering from Cleveland State University.

Xilinx

2100 Logic Drive
San Jose, CA 95124

Tel: 408-559-7778

Email: tom.hill@xilinx.com

Website: www.xilinx.com